Theorem

The greedy algorithm to compute a superstring is a 3-approximation algorithm for MCCSP.

Proof:

Greedy: $(S_{j_1}, S_{j_2}, \ldots, S_{j_n})$

Optimum: $(S_{i_1}, S_{i_2}, \ldots, S_{i_n})$

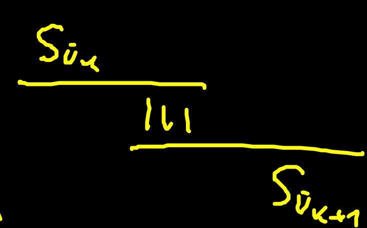
Wir zeigen: $\langle S_{j_k}, S_{j_{k+1}} \rangle$ kann höchstens drei Merges der optimalen Lösung unmöglich machen:

1. Fall: $\langle S_{j_k}, S_{j_{k+1}} \rangle = \langle S_{i_l}, S_{i_{l+1}} \rangle$ für $l \in [1:n-1]$

$\rightarrow$ beide identisch, kein Merge verhindert

2. Fall: $\langle S_{j_k}, S_{j_{k+1}} \rangle = \langle S_{i_l}, S_{i_{l+m}} \rangle$, $m > 1$, l geeignet

$\rightarrow$ höchstens 2 Merges unmöglich, nämlich



$\langle S_{i_l}, S_{i_{l+1}} \rangle$ und

$\langle S_{i_{l+m-1}}, S_{i_{l+m}} \rangle$

3. Fall: $\langle S_{j_k}, S_{j_{k+1}} \rangle = \langle S_{i_{l+m}}, S_{i_l} \rangle$ für geeignetes l und $m > 0$

$\rightarrow$ höchstens 3 Merges unmöglich

$\langle S_{i_{l+m}}, S_{i_{l+m+1}}\rangle$, $\langle S_{i_{l-1}}, S_{i_l}\rangle$ und einer der Merges $\langle S_{i_l}, S_{i_{l+1}}\rangle, \dots\dots, \langle S_{i_{l+m-1}}, S_{i_{l+m}}\rangle$, da Anwendung zu Zykel führte

$$\begin{tikzcd} S_{i_1} \arrow[r] & S_{i_2} \arrow[r] & \cdots \arrow[r] & S_{i_l} \arrow[r, red] & \cdots \arrow[r, red] & S_{i_{l+m}} \arrow[r] \arrow[ll, bend left, blue] & \cdots \arrow[r] & S_{i_n} \end{tikzcd}$$

$\leftarrow$ schließt Zykel $\Rightarrow$ eine der Kanten $\rightarrow$ unmöglich.

$M_x \mathrel{\hat{=}}$ Menge der Merges von Algorithmus $x \in \{g, o\}$

$V(m) \subseteq M_o \mathrel{\hat{=}}$ Menge der von $m = \langle S_{j_k}, S_{j_{k+1}}\rangle \in M_g$ verhinderten Merges

Für $ov(m)$, $m \in M_g$, Länge des durch m verschmolzenen Overlaps, gilt:

$$(\forall m \in M_g): \left(ov(m) \geq \frac{1}{3} \left(\sum_{m' \in V(m)} ov(m') \right) \right) \qquad (*)$$

$\uparrow$ max. 3 Summanden

denn Greedy wählt unter verbl. Merges immer den mit max. Overlap $\rightarrow$ von m verhinderter Merge hat nie Overlap $>$ $ov(m)$.

Außerdem gilt

$$(\forall m' \in M_o): (m' \in M_g) \vee ((\exists m \in M_g \setminus \underline{M_o}): (m' \in V(m)))$$

(**)

denn für $m' = \langle s, \bar{s} \rangle \in M_g$ muß es in M_g Merges geben, die s und $\bar{s}$ mit anderen Wörtern verschmelzen, wodurch m' verhindert wird. Ein solcher Merge kann nicht in M_o liegen, da er ja $m' \in M_o$ verhindert.

Betrachte nun

$$\sum_{m \in M_g} ov(m) = \sum_{m \in M_g \cap M_o} ov(m) + \sum_{m \in M_g \setminus M_o} ov(m)$$

Es gilt:

$$\sum_{m \in M_g \cap M_o} ov(m) \geq \frac{1}{3} \sum_{m \in M_o \cap M_g} ov(m) \qquad \text{und}$$

$$\sum_{\substack{m \in M_g \setminus M_0 \\ \subseteq M_g}} ov(m) \overset{(*)}{\geq} \sum_{m \in M_g \setminus M_0} \frac{1}{3} \sum_{m' \in V(m)} ov(m')$$

$$\overset{(**)}{\geq} \frac{1}{3} \sum_{m' \in M_0 \setminus M_g \subseteq M_0} ov(m')$$

$$\Longrightarrow \frac{comp(w_0)}{comp(w_g)} = \frac{\sum_{m' \in M_0} ov(m')}{\sum_{m \in M_g} ov(m)} \leq \frac{\sum_{m' \in M_0} ov(m)}{\frac{1}{3} \sum_{m' \in M_0} ov(m')} = 3 \qquad \square$$

Using more elaborate reasoning one can show:

Theorem

The greedy algorithm to compute a superstring is a 2-approximation algorithm for MCCSP. □

Alternative Procedure

Idea: Do not use a single cycle to cover all vertices but a set of cycles

$\rightsquigarrow$ cycle-cover for which each vertex must be part of exactly one cycle.

Here: Cycle-cover of minimal cost (according to distance graph).

Remark: A minimal cycle-cover can be computed in time $\mathcal{O}(n^3)$ (whereas the restriction to edges to be covered by at least one cycle leads to an $\mathcal{NP}$-complete problem).

Def.: Sei $G = (V, E, g)$ ein vollständiger Digraph mit Kantengewichten $g: E \to \mathbb{N}$. Ein Cycle-Cover von

G besteht aus einer Menge $\mathcal{K} = \{K_1, \dots, K_e\}$ gerichtete Kreise in G, so dass jeder Knoten in V in <u>genau</u> einem Kreis $K_i \in \mathcal{K}$ vorkommt.

Kosten: $\sum$ Kantengewichte aller Kreise.

Approxiamtion: Construct a single cycle (solution to TSP) from the minimal cycle-cover.

Steps:

1. Identify each cycle $c \in \mathcal{C}$ by one of its vertices $\rightsquigarrow$ set of vertices R.
2. Compute a minimal cycle-cover for the subgraph induced by $R \rightsquigarrow$ set of cycles $\mathcal{C}'$.
3. For each $c \in \mathcal{C}'$ delete the edge corresponding to a merge of minimal overlap $\rightsquigarrow$ broken cycles are considered as merged strings.
4. Concatenate these strings and expand each of its elements by the cycle from $\mathcal{C}$ it represents.

$K_1 = a_1 \to a_2 \to \dots \to a_k \to a_1 = \langle a_1, a_2, \dots, a_1 \rangle$

$K_2 =$

$\vdots$

$K_7 = g_1 \to g_2 \to g_3 \to g_1$

Repräsentant r_{K_i}

alle zusammen = R

Cover für r_{K_i}:

$K_1' = a_1 \cancel{\to} b_1 \to c_1 \to a_1$

$K_2' = d_1 \cancel{\to} e_1 \to d_1$

$u_3' = f_1 \to g_1 \to f_1$

$$\left.\begin{array}{l} u_1 = \langle b_1, c_1, a_1 \rangle \\ u_2 = \langle e_1, d_1 \rangle \\ u_3 = \langle g_1, f_1 \rangle \end{array}\right\} \rightsquigarrow w' = u_1 u_2 u_3$$

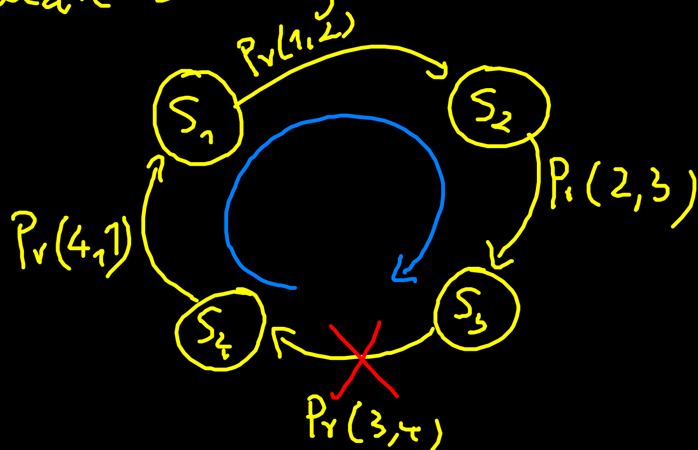
In w' ersetzen wir die Repräsentanten durch die zu ihm geh. „Kreise"

Theorem

The algorithm just outlined is a 3-approximation algorithm for SCSP.

Proof:

Optimale Lösung (TSP) sei



Kosten TSP: $Pr(1,2) + Pr(2,3) + P(3,4) + Pr(4,1) \longrightarrow$ Kosten für SCSP $Pr(4,1) + Pr(1,2) + Pr(2,3) + |S_3|$

Es gilt: $|S_i| > Pr(i,j) \Longrightarrow$ Kosten TSP stets untere Schranke für SCSP. Außerdem sind Kosten einer min. Cycle-Cover untere Schranke für Kosten TSP

$$\Rightarrow \quad cost(\mathcal{K}) = Opt_{cc}(DG(S)) \leq Opt_{scs}(S)$$

$$cost(\mathcal{K}') = Opt_{cc}(DG(R)) \leq Opt_{scs}(R)$$

Außerdem $cost(\mathcal{K}') \leq cost(\mathcal{K})$ (o)

$$|u_i| = cost(K_i') + minov(K_i')$$

min. Overlap des Kreises K_i'

$$|w'| \leq cost(\mathcal{K}') + \sum_{k' \in \mathcal{K}'} minov(k') \quad (*)$$

Ersetzen wir in w' alle Repräsentanten durch „Kreise"

$$\rightsquigarrow \quad r_{k_1} = a_1 \rightsquigarrow \underbrace{P_1(1,2) \cdot P_1(2,3) \cdots P_r(4,1)}_{\text{Gewicht}} \cdot a_1$$

w' verlängert sich um Gewicht

$$|w| \leq |w'| + cost(\mathcal{K})$$

$$\overset{(*)}{\leq} cost(\mathcal{K}') + \sum_{k' \in \mathcal{K}'} minov(k') + cost(\mathcal{K})$$

$$\overset{(o)}{\leq} 2 \cdot cost(\mathcal{K}) + \sum_{k' \in \mathcal{K}'} minov(k')$$

Jeder Kreis mind.

Nun gilt offensichtlich

2 Kanten

$$\sum_{K' \in \mathcal{K}'} \text{minov}(K') \leq \frac{1}{2} \cdot \sum_{K' \in \mathcal{K}'} \sum_{(s_i, s_j) \in K'} ov(s_i, s_j)$$

$$\Longrightarrow |w| \leq 2 \cdot cost(\mathcal{K}) + \frac{1}{2} \sum_{K' \in \mathcal{K}'} \sum_{(s_i, s_j) \in K'} ov(s_i, s_j)$$

<u>Lemma:</u> K_1, K_2 zwei Kreise in min. CC $\mathcal{K}$ und $w_1 \in K_1$, $w_2 \in K_2$ zwei Wörter in diesen Kreisen. Dann gilt

$$ov(w_1, w_2) < cost(K_1) + cost(K_2)$$

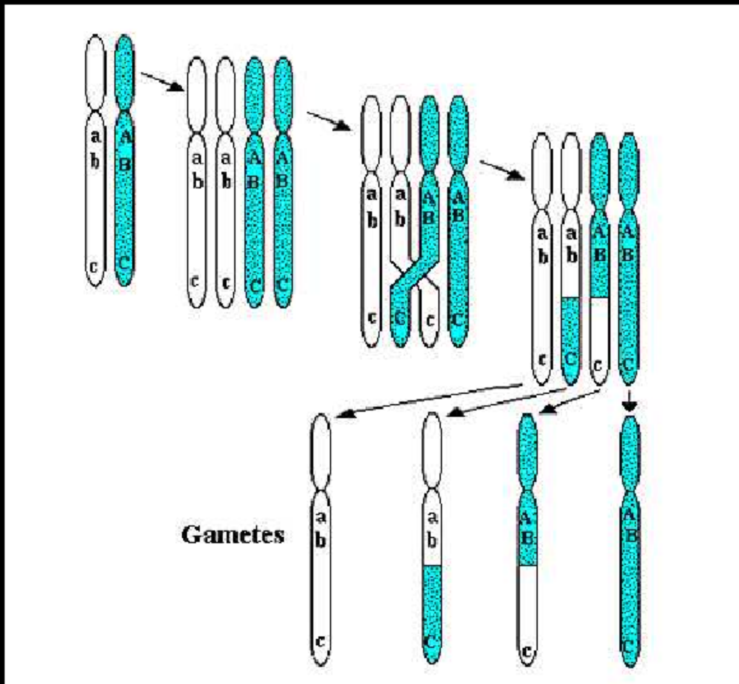
$$\Longrightarrow \sum_{K' \in \mathcal{K}'} \sum_{(s_i, s_j) \in K'} ov(s_i, s_j) < \sum_{K' \in \mathcal{K}'} \sum_{(s_i, s_j) \in K'} \left(cost(K_i) + cost(K_j) \right)$$

$$= 2 \cdot cost(\mathcal{K})$$

$$\Longrightarrow |w| \leq 3 \cdot cost(\mathcal{K})$$

$$\leq 3 \cdot Opt_{SCS}(S)$$

$\square$



Crossing-over and recombination during meiosis